

AUTOMATIC MUCK PILE CHARACTERIZATION FROM UAV IMAGES

Fabian Schenk¹, Alexander Tscharf², Gerhard Mayer², Friedrich Fraundorfer¹

¹ Institute of Computer Graphics and Vision, Graz University of Technology, Austria
(schenk, fraundorfer)@icg.tugraz.at

² Chair of Mining Engineering and Mineral Economics, Montanuniversität Leoben, Austria
(alexander.tscharf, gerhard.mayer)@unileoben.ac.at

KEY WORDS: Muck pile characterization, Fragment size distribution, Mining, Machine-learning, UAV, Computer Vision

ABSTRACT:

In open pit mining it is essential for processing and production scheduling to receive fast and accurate information about the fragmentation of a muck pile after a blast. In this work, we propose a novel machine-learning method that characterizes the muck pile directly from UAV images. In contrast to state-of-the-art approaches, that require heavy user interaction, expert knowledge and careful threshold settings, our method works fully automatically. We compute segmentation masks, bounding boxes and confidence values for each individual fragment in the muck pile on multiple scales to generate a globally consistent segmentation. Additionally, we recorded lab and real-world images to generate our own dataset for training the network. Our method shows very promising quantitative and qualitative results in all our experiments. Further, the results clearly indicate that our method generalizes to previously unseen data.

1. INTRODUCTION

Open pit mine and quarry blasts result in a wide-spread muck pile containing thousands of individual stone fragments. The results of a production blast may be described in terms of the fragmentation and the geometric properties of the individual fragments, such as shape, angularity or roundness. Fast information about the muck pile properties are essential for production scheduling and further industrial processing and additionally this information can be used to optimize blasting patterns of future blasts.

In mining, a widely used and well-known representation of muck pile characteristics is as cumulative size distribution (CDF, or sieving curve), which gives a complete description of the fragmentation. The CDF represents the "fraction of mass P passing a screen with a given mesh size x " and $P(x)$ is in the range of 0-100% (see Fig. 1 (3)). From a practical point of view one may describe a number of relevant quantities, whereas the x_{50} value to measure the average fragmentation is the most important (Schubert, 1989). It describes the mesh size x_{50} through which half of the muck-pile passes.

To compute CDF curve, it is necessary to determine the size of each individual fragment, which is why sieving constitutes the only or at least most common "true fragment size distribution", if the process itself is done error free. Since the determination of the characteristics of a modern production blast by means of screening analysis is neither practical nor economically feasible, image-based methods such as WipFrag, FragScan, Split-Desktop, Power-Sieve, IPACS, TUCIPS, CIAS and GoldSize have become increasingly popular in recent years. However, the definition of image based fragment size is different to screening analysis and the results of existing solutions are strongly depending on user intervention, expert knowledge and the form of image acquisition (Latham et al., 2003, Tscharf et al., 2018). With regards to image acquisition, modern unmanned aerial vehicles (UAVs) are capable of recording images of a wide blast area within minutes at very high resolution (see Fig. 1 (1)). Especially in hazardous mining environments they help to overcome geometric constraints and avoid visibility problems.

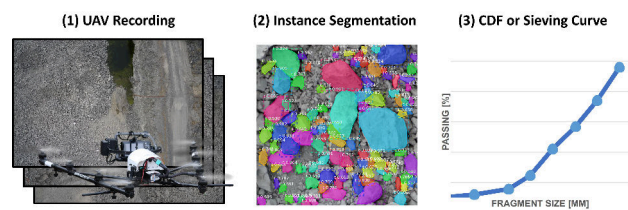


Figure 1. (1) A drone records images after a blast, then (2) our method segments individual fragments from these images to (3) automatically generate the CDF or sieving curve.

From these UAV recordings the individual fragments have to be identified (see Fig. 1 (2)). In computer vision, this task is typically referred to as instance segmentation, where all objects have to be correctly detected and precisely segmented. It is a combination of objection detection, where individual objects have to be classified and localized, and semantic segmentation, where each pixel is classified into a fixed set of categories without distinguishing between individual instances.

With traditional delineation methods such as edge detectors or superpixel segmentation (see Fig. 2 (a) and (b)), it is difficult to find the borders between fragments and a non-trivial post-processing step is necessary to distinguish between individual fragments and background. Additionally, shading and partial or complete overlapping pose a major problem.

In recent years, nearly all fields of computer vision have rapidly progressed due to heavy use machine-learning methods with deep convolutions neural networks (CNNs) (Krizhevsky et al., 2012, LeCun et al., 1989) being the most popular. The main advantage of machine-learning methods is that given enough training data, they can cope with the high variability in realistic scenes without relying on user intervention or scene-specific thresholds. Several approaches to tackle the challenge of instance segmentation have been proposed in the last years, but the most promising is Mask R-CNN from (He et al., 2017), which simultaneously predicts a bounding box, a confidence value and a precise segmentation mask, while running relatively fast on a GPU.

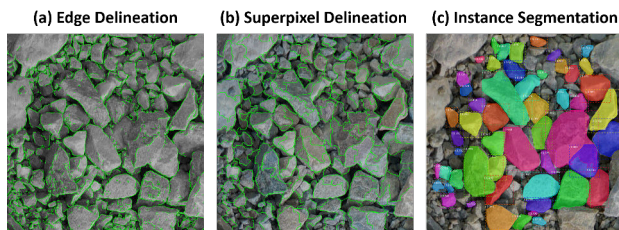


Figure 2. (a) Edge delineation does not detect closed boundaries, while (b) superpixel segmentation often splits larger fragments into smaller ones. (c) Our method successfully segments individual fragments.

In this work, we propose a novel method for muck pile characterization based on the segmentation of individual stone fragments from UAV images after a mine blast (see Fig. 1). Our method builds upon Mask R-CNN and works fully automatically without any manual threshold settings or user intervention. The main contributions can be summarized as follows:

- Creation of a lab and a real-world UAV dataset
- A multi-scale instance segmentation algorithm
- A method to fuse all segmentations into one globally consistent one.
- Computation of the fragment sizes and CDF curve
- Extensive qualitative and quantitative experiments
- Comparison to a state-of-the-art commercial software

2. RELATED WORK

We will first discuss commercially available image-based muck pile characterization software and then continue with a review of machine-learning based instance segmentation methods.

2.1 Digital Image analysis for determination of CDF

To determine the fragment size distribution of a full-scale production blast (some tens of thousands of tons of material) through sieving is an exceedingly time consuming and expensive task. Therefore it is not surprising that photographs emerged as an alternative, even though there are - on top of representation errors - several new tasks to cope with: (1) What do the images show? (flat lying blocks in contrast to the consideration of the smallest, mostly set up cross section with the screening analysis). (2) How to evaluate the 2D fragment size distribution in the image and how to transform it into volume or masses of fragments? (3) How large are the finest particles to be detected? (4) How to treat large blocks that are covered or lie only partly in the image?

Many commercial solutions like WipFrag, PowerSieve or Split-Desktop try to solve these tasks using classical image processing approaches for particle delineation, but accuracy, reproducibility and robustness is not always clear (Sanchidrián et al., 2009, Thurlay, 2014). Basically all are showing two common weaknesses: First the inability to robustly resolve more than 1-1.5 orders of magnitude in fragment size, which is a serious problem especially for blasted muck piles, where the fragments cover at least 3 orders of magnitude. And second the tendency to give steeper or

more uniform fragmentation curves than sieving, which is mainly due to errors in the extraction of individual fragments. Large particles tend to "disintegrate" into several parts and smaller ones are "merged" into larger ones, even in systems that do not employ edge detection techniques for segmentation (see Fig. 7 and Fig. 8).

Investigations on the accuracy of image-based methods show that while image analysis methods function relatively well in the range of large grain sizes (error < 30%), reliable results cannot be achieved in the fine range (Sanchidrián et al., 2009). "Fines correction" increases the quality of the results as it is shown by (Maerz and Zhou, 2000), who present three methods which are implemented in WipFrag system: (1) "Analytical correction", which compensates for missing fines by considering the smaller probability of a finer fragment being detected in a sampling plane. (2) "Zoom-Merge correction", which involves acquiring numerous images at different resolution levels and merging them in the final analysis. (3) "Rosin-Rammler Empirical Calibration Correction", which follows the assumption that for a given comminution process the shape of the distribution is more or less constant, whereas the parameters of the distribution function (e.g. Rosin-Rammler curve (Schubert, 1989)) can be determined either by screening in real scale or using model studies. Split-Desktop does fines correction, by estimating the remaining finer material below a certain fines cutoff that depends on the resolution of the image. The shape of the curve below this cutoff is determined by the distribution of the particles above the cutoff. Split-Desktop utilizes the best fit of either a Schumann or Rosin-Rammler equation to represent the distribution of fines below the automatically computed fines cutoff point (Split Engineering LLC, 2016). Further, SplitDesktop does not distinguish between fore- and background but rather assumes that all visible surfaces are fragments (see Fig. 7 and 8)

In summary, all existing solutions need some kind of previous knowledge about the underlying distribution function or at least strong user interaction defining the relevant threshold values or editing the automated delineations. The achievable results are therefore strongly dependent on the experience of the user, and are only conditionally suited to characterize blasted muck piles.

2.2 Instance Segmentation

One of the most challenging tasks in computer vision is instance segmentation, which requires detection and localization of individual objects, while also precisely segmenting each instance.

In the past years, region-based CNNs (Girshick et al., 2014, Girshick, 2015) for object bounding box detection have been at the core of many instance segmentation methods. The main idea is to keep a manageable number of candidate object regions and process them individually with a CNN. DeepMask (Pinheiro et al., 2015) and SharpMask (Pinheiro et al., 2016) learn to predict segment proposals that are then classified by Fast R-CNN (Girshick, 2015). (Dai et al., 2016) apply a similar concept in their three-step multi-stage cascade system. They (i) generate bounding box proposals (ii) that are then segmented and (iii) finally classified. All these approaches are difficult to train and slow due to their complex structure.

(Yi Li and Wei, 2017) introduced the "fully convolutional instance segmentation" FCIS, where they combine segment proposals and objection detection to fully convolutionally predict a

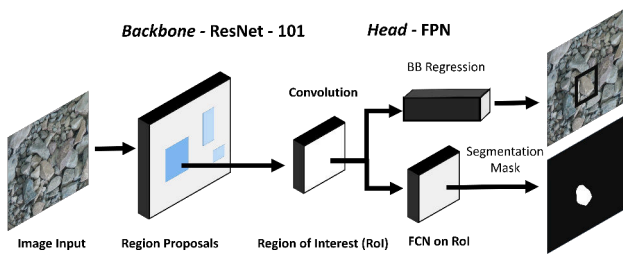


Figure 3. The network first predicts region proposals for an input image. Then for each region, a separate BB and binary segmentation mask is predicted. The network consists of a *backbone* for feature extraction and a fully convolutional *head* part.

set of position-sensitive output channels. These channels simultaneously address object classes, bounding boxes and masks. However, FCIS has problems with overlapping instances since these instances “compete” for the final segmentation.

The state-of-the-art algorithm in this field is Mask R-CNN proposed by (He et al., 2017). In this work, we build upon Mask R-CNN since it is faster than DeepMask, more flexible and simpler than the approach by (Dai et al., 2016) and does not show errors for overlapping instances. Since Mask R-CNN is an integral part of our method, we will explain it in detail in Section 3.1.

3. IMAGE-BASED MUCK PILE CHARACTERIZATION

We present a novel fully automatic method to address the challenge of image-based muck pile characterization. Figure 1 shows an overview of our method, where we (1) process an input image, (2) segment individual stone fragments and (3) determine the smallest diameter of each fragment to compute the CDF or sieving curve. At the core of our method is a modified version of Mask R-CNN to detect the individual fragments. In this chapter, we will first introduce the network architecture and show how to train the network and then we will describe the other parts of our method.

3.1 Network Architecture

The basis of our method is the state-of-the-art instance segmentation Mask R-CNN (He et al., 2017). One major difference is that instead of 81 classes we only predict two classes: *fragment* and *background*. Similar to its predecessor Faster R-CNN (Girshick, 2015), Mask R-CNN is also a two stage algorithm that initially proposes candidate object bounding boxes (BB) with a Region Proposal Network (see Fig. 3).

In contrast to Faster R-CNN, Mask R-CNN additionally predicts separate binary segmentation masks by applying a fully convolutional network (FCN) to each region proposal. As depicted in Figure 3, Mask R-CNN algorithm can be divided into two different parts: (i) the *backbone* architecture that is responsible for feature extraction over an entire image and (ii) the *head* part for BB recognition and mask prediction. Figure 3 shows an overview of the network architecture, where we first extract a feature map from the input image and predict region proposals for which we then separately predict BB and segmentation mask. Even though Mask R-CNN is a meta algorithm and does not depend on a specific architecture, we follow the suggestion of the authors and use a Feature Pyramid Network (FPN) (Lin et al., 2017) as *backbone* and ResNet 101 (He et al., 2016) as *head*. One of the main

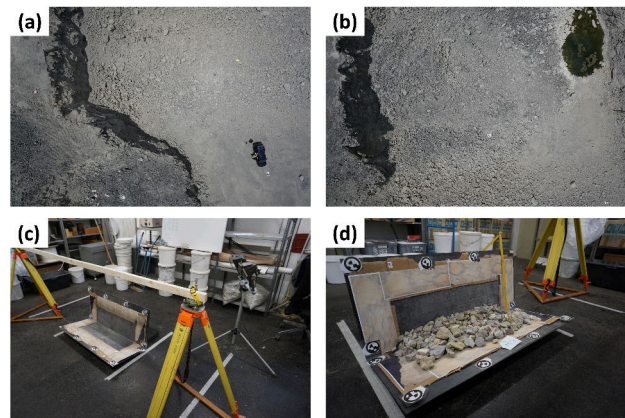


Figure 4. (a, b) UAV recordings from an open pit mine at El Ajibe, Spain, at a height of 30 m. The recording setup in the lab with (d) and without (c) a muck pile.

benefits of Mask R-CNN is that it decouples mask and class/BB prediction, since it predicts a binary mask for each class independently, avoiding the problem of competing classes or instances that other methods such as FCIS suffer from.

3.2 Network Training

A large amount of training data is what drives modern machine-learning methods and like all supervised methods, also ours relies on a vast amount of training data. However, the annotation of thousands of images is a tedious and very time-consuming task and training a complete network from scratch takes many days up to weeks even on multiple GPUs. Therefore, a CNN is typically initialized with pre-trained weights to provide a good starting point and then it is fine-tuned for a specific task. To the best of our knowledge, for muck pile characterization no annotated dataset is available. Thus, we create our own dataset consisting of recordings from a lab setup, where the CDF of the muck pile is known, as well as many UAV recordings from a real open pit mine.

3.2.1 Dataset Recording We started by recording the whole blasting cycle of production rounds with a DJI Matrice 600 at an open pit mine at El Ajibe, Spain. The DJI Matrice 600 was equipped with Sony 7R camera with 25mm fixed lens and a resolution of 42.4Mp to capture nadir images with an overlap of 90% at a flying altitude of 16 and 30m (see Fig. 4 (a) and (b)). In a total, we performed 23 UAV flights and recorded around 400 images per flight.

Especially in the development phase and for evaluation of our system, ground truth information on the characteristics of the muck pile is of high importance. As it is not possible to do screening analysis of the whole muck pile in real scale, we designed a lab scale experiment with known fragment size distribution, which is suited for training the network as well as for evaluation of the results. Figure 4 (c) shows the recording setup for the lab scale experiments.

In the lab, we use rock material from a real mine blast to set up several synthetic muck-piles with different fragment sizes and compositions at a scale of 1:25 (see Fig. 4 (d)). In total, we tested four different fragmentation distributions: uniform fragmentation (**Dist 1**), fine material (**Dist 2**), coarse (**Dist 3**) and a dust and boulders fragmentation (**Dist 4**), where each distribution was

arranged in at least four different ways. We record several images of each arrangement from multiple perspectives under different lighting conditions to simulate the different angles seen by a drone. Since the CDF of each distribution is known, it serves as ground truth for the experiments presented in Section 4.1.

3.2.2 Dataset Preparation We generate training images for the network by annotating around 1000 individual fragments with the VIA annotation tool (Dutta et al., 2016). Due to the limited camera resolution, we do not annotate fragments under 10 mm since the exact boundaries of very such small fragments are difficult to distinguish even for a human. However, we address this issue at later stage with a multi-scale segmentation process (see Sec. 3.3).

State-of-the-art CNNs have millions of different parameters and fine-tuning requires a proportional high amount of ground truth data. Since annotating fragments is a tedious and time-consuming task, in practice it is not possible to provide ground truth labels for millions of individual fragments. Thus, we follow the common practice and enlarge the dataset by a factor of 10 through standard augmentation techniques such as mirroring, rotating and cropping followed by up-/down-scaling during training. The augmentations are generated on-the-fly during training and do not require memory on the hard drive or other pre-processing steps.

3.2.3 Training Procedure We initialize the network with the pre-trained weights from Mask R-CNN trained on several thousand images of the MS COCO dataset (Lin et al., 2014) and focus only on fine-tuning the network. We minimize a loss L :

$$L = L_{cls} + L_{bb} + L_{mask}, \quad (1)$$

where L_{cls} and L_{bb} are the BB and class loss identical to (Girshick, 2015) and L_{mask} is the binary cross-entropy loss. Since we only fine-tune the network, we refer the interested reader to (He et al., 2017) for details about training from scratch.

We use a multi-step training procedure, where we first fine-tune only the head architecture for 5 epochs and then fine-tune both, *head* and *backbone* over 10 epochs. We crop a total of 4323 patches from various distributions and split them into a training and a validation set at a ratio of 95% to 5%. Note that each cropped patch must contain at least one fragment. We then train the network with a learning rate of $l = 0.0001$, a non-max suppression threshold of 0.2 and a minimum confidence value for a segmentation to be accepted of 0.45.

3.3 Fragment Instance Segmentation

Our method segments individual fragments in an image and simultaneously predicts a confidence score, a bounding box and binary segmentation mask. The maximum image resolution r_{net} that can be passed through the network is limited by the available GPU memory and in our case is $r_{net} = [1024 \times 1024]$. UAV images are typically in the range of 30 and 40 megapixels and too large to be processed in one pass. Thus, we crop different patches of size r_{net} with an overlap of $\frac{r_{net}}{2}$ from the image and process them individually. The borders of the cropped patches are also problematic, since they often contain only parts of a fragment. In order to avoid partly segmented fragments that would comprise the computed CDF, we remove objects whose BB are closer than 10 *px* to the border of patch. For the final segmentation, we do not consider objects with a confidence score lower than 0.45.

As mentioned in Section 3.2.2, the training data lacks annotation of small fragments. We address this problem with a multi-scale

segmentation approach that is capable to detect such small fragments. All the segmentations are then fused into a final global segmentation.

3.3.1 Multi-Scale Processing Instead of processing only image patches with resolution r_{net} , we crop patches with a lower resolution r_{patch} , e.g. $r_{patch} = \frac{r_{net}}{2}$, and bilinearly upsample the patch to r_{net} for processing. For the lower resolution patches, we keep an overlap of $\frac{r_{patch}}{2}$. The results are then scaled back to the original resolution and fused into the global segmentation.

Figure 6 shows the difference between the number of segmented fragments at a patch size of $r_{patch} = 1024 \times 1024$ when processing only a single-scale, which corresponds to a scaling factor of $F = [1]$ compared to multi-scale processing at half 2 and 3 different scale levels. It is clearly visible that the multi-segmentation approach is capable of segmenting smaller fragments even though no annotations are present in the dataset. The number of individual fragments also greatly increases, enabling a much more reliable computation of the CDF.

3.4 Segmentation Fusion

Each patch consists of many individual fragment segmentations that are typically present also in other patches. A globally consistent segmentation is essential for an accurate computation of the CDF.

We propose a method to fuse multiple segmentations over various scale levels into one globally consistent segmentation mask. We always keep a set of global segmentation masks \mathcal{S} , their corresponding confidence values and BBs. We fuse the set of segmentation masks \mathcal{S}_p , corresponding confidence values and BBs of every patch individually into the global segmentation. For the individual masks $\mathcal{S}_{p,i} \in \mathcal{S}_p$ of a patch, we compute the overlap with \mathcal{S} as:

$$O_{ij} = \mathcal{S}_{p,i} \cap \mathcal{S}_j \quad \forall \mathcal{S}_j \in \mathcal{S}. \quad (2)$$

Since overlaps are expensive to compute, we avoid unnecessary computations and focus only on the masks with overlapping BBs. However, overlapping BBs do not guarantee an overlap of the segmentation masks, e.g. the BB of the green fragment in Fig. 6 is much larger than the actual mask. If there is no overlap with \mathcal{S} , we simply add $\mathcal{S}_{p,i}$ to the global segmentation \mathcal{S} . Otherwise, we compute the relative overlap p_i and p_j for each mask in pixels as:

$$p_i = \frac{O_{ij}}{|\mathcal{S}_i|}, \quad p_j = \frac{O_{ij}}{|\mathcal{S}_j|} \quad (3)$$

where $|\mathcal{S}_i|$ and $|\mathcal{S}_j|$ represent the segmented area of the masks. Depending on p_i and p_j , our proposed fusion method distinguishes between the following two cases:

1. $p_i < \Theta_S$ and $p_j < \Theta_S$, i.e. a very small overlap, we again add the segmentation to the global mask \mathcal{S} .
2. $p_i \geq \Theta_S$ and $p_j \geq \Theta_S$, i.e. probably the same detection, we keep the one with the highest confidence.
3. All other cases, e.g. a small fragment on top of another, currently do not contribute to the global segmentation

During our experiments, we found that a value of $\Theta_S = 50\%$ is a suitable choice.

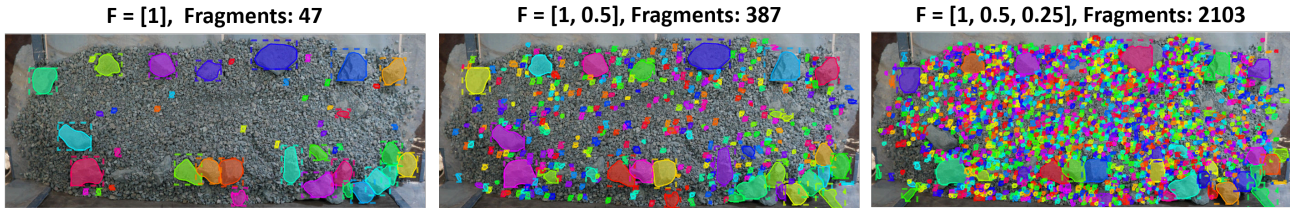


Figure 5. Comparison of the multi-scale segmentation on **Dist 4** with 1, 2 and 3 scale levels indicated by the scale factors $F = [1, 0.5, 0.25]$, which correspond to resolutions of $r_{patch} = [1024 \times 1024], [512 \times 512], [256 \times 256]$.

3.5 Fragment Size Distribution Computation

In Section 1, we introduced the CDF, which visualizes the distribution as a single curve and is far more compact and more easily conceivable than an image of segmented fragments. Sieving is done in several steps by allowing the material to pass through a series of sieves of progressively smaller mesh size and weighing the amount of material that is stopped by each sieve as a fraction of the whole mass. The curve is then computed as cumulative sum of each bin and presented in a graph of percent passing versus the sieve size in logarithmic scale (see Fig. 7). For the sieving curve, only the smallest possible width of a fragment is relevant. In order to generate a sieving curve out of segmented image data, we propose two different ways to compute the minimum diameter of each individual fragment: (i) approximation as a circle and (ii) approximation as an ellipse.

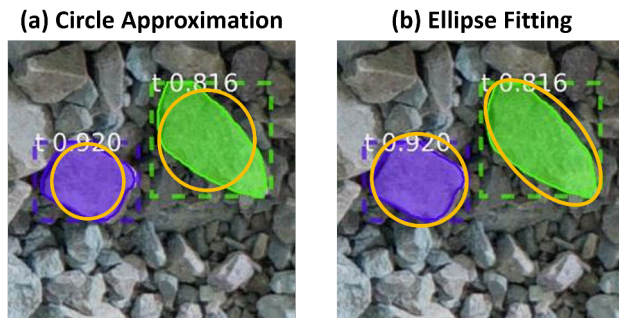


Figure 6. (a) the approximation of the area of the fragment as circle and computing the diameter, where especially elongated objects are misrepresented. In comparison, (b) shows that a fitted ellipse represents elongated objects more accurately.

Approximation as a Circle: Many fragments have a circular shape and it is intuitive to approximate them as circle. The smallest diameter d_{circle} of a fragment with size A is then given as:

$$d_{circle} = 2\sqrt{\frac{A}{\pi}}. \quad (4)$$

Figure 6 (a) shows the computed circular shape of two fragments. However, for the fragment size distribution, only the smallest diameter is relevant and in the circular approximation elongated objects appear wider than they actually are, e.g. the green fragment in Figure 6 (a). In general, the approximation as circle results in the computed fragment sizes being larger than they actually are, which shifts the CDF curve to the right.

Approximate as an Ellipse: We tackle this problem by fitting an ellipse instead of a circle around the fragment, which describes elongated objects much more accurately (see Fig. 6(b)). To fit an ellipse, we first compute the contours similar to (Suzuki et al., 1985) and then apply the direct ellipse fitting algorithm proposed

x_{50}	Dist 1		Dist 2		Dist 3		Dist 4	
Method	A_1	A_2	A_1	A_2	A_1	A_2	A_1	A_2
GT	14 - 20		6 - 10		31.5 - 40		6 - 10	
SD	18.4	19.3	7.5	8.9	36.1	31.7	20.8	14.8
Circle	42.6	44.5	22.0	22.0	45.9	51.2	57.4	59.6
Ellipse	36.4	32.7	18.2	17.9	37.8	41.6	49.5	52.0
MSCir	29.0	28.2	16.7	17.1	41.8	48.73	9.8	9.7
MSEll	24.8	23.5	13.6	14.2	34.8	39.1	8.0	8.1

Table 1. Comparison of the computed x_{50} values in [mm] on two arrangements A_1 and A_2 of four different distributions. Values within the GT range are in bold.

by (Fitzgibbon et al., 1999). The smallest diameter of the fragment defines its size can be computed from the two ellipse axes a_1 and a_2 as:

$$d_{ellipse} = \min(a_1, a_2). \quad (5)$$

In the results section we show that this method does not have a bias towards over-estimating the fragment sizes.

3.6 Implementation

Training and inference run on a desktop PC with an Intel Core i7-4790 @ 3.60 GHz x 8 cores with 32 Gb of main memory and an Nvidia GeForce GTX 970 graphics card with 4 Gb RAM. We extend the Matterport implementation¹ of Mask-RCNN. As third-party libraries, we utilize Keras, Tensorflow, Python, Cuda and the pyCocoTools.

4. RESULTS AND DISCUSSION

In this section, we compare the commercial Split-Desktop (SD) (Split Engineering LLC, 2016) software to four different versions of our method:

- Single scale with circle approximation (Ours-Circle)
- Single scale with ellipse approximation (Ours-Ellipse)
- Multi-scale with circle approximation (Ours-CircleMS)
- Multi-scale with ellipse approximation (Ours-EllipseMS)

The single scale approaches run with a resolution of $r_{patch} = [1024 \times 1024]$, while the multi-scale approaches run at 3 levels with resolutions of $[1024 \times 1024]$, $[512 \times 512]$ and $[256 \times 256]$. The comparison with the commercial and state-of-the-art solution Split-Desktop was carried out as blind comparison on the same images but without previous knowledge about the underlying distribution function. The delineation was qualitatively adjusted by using the "level of delineation" slider, but not manually

¹https://github.com/matterport/Mask_RCNN

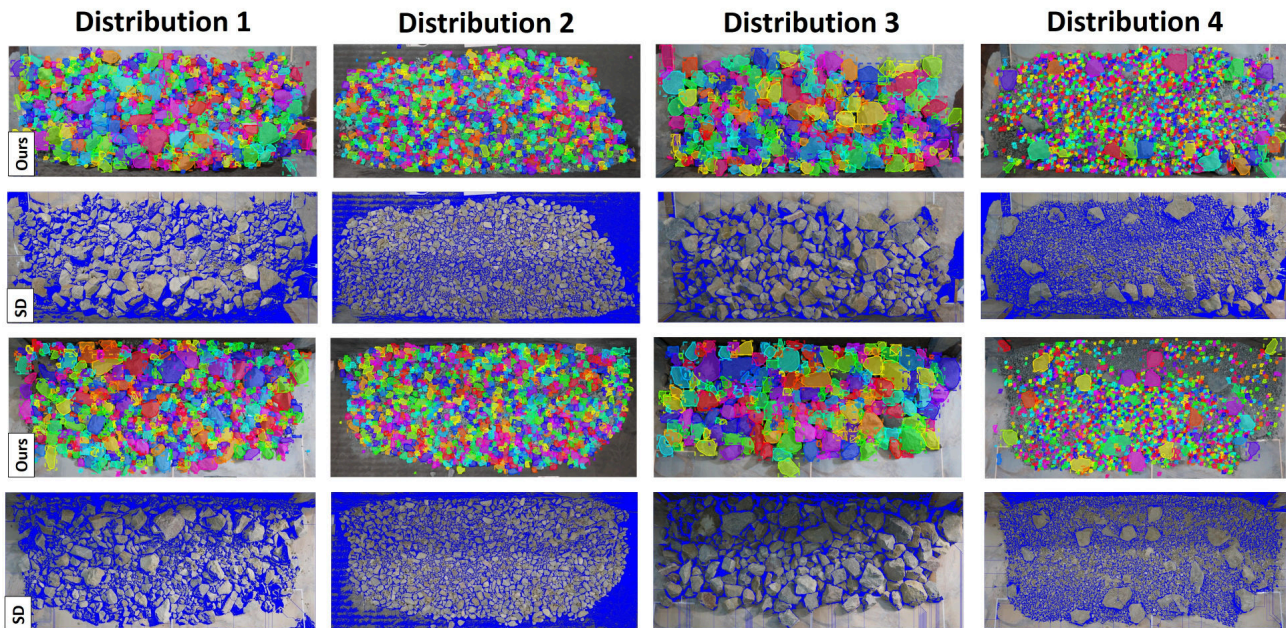


Figure 7. Qualitative results of our MSELLipse method compared to SD for all four distributions and arrangements.

edited to ensure a fair comparison as manual editing is not done in our automated approach. Additionally the "boulder detection" was used, which optimizes results with special regard to not split larger fragments. Finally the fines factor, which is a powerful user controlled parameter strongly affecting the results, was adjusted separately for every image. It represents the percentage of blue pixels, which are considered as fines, as those pixels could represent both outlines of particles and fines.

We show quantitative results on the lab dataset with known CDF in the form of the x_{50} value, which is representing the average fragmentation, i.e. the mesh size through which half of the muck pile passes. Additionally, we present qualitative results on our lab dataset and one real-world UAV recordings of both our method and SD.

4.1 Evaluation on the Lab Dataset

In this section, we present the quantitative and qualitative results on the lab dataset. From each of the 4 distributions, we select 2 different arrangements A_1 and A_2 that the network has never seen during training. Qualitatively, we evaluate the x_{50} value and compare it to the ground truth (GT). While our method and SD can compute an exact x_{50} value, from the ground truth screening analysis we only know a certain range in which the x_{50} lies, i.e. the mesh size till which 50% passed. For the sieving analysis, we use mesh sizes [3.15, 6, 10, 14, 20, 25, 40, 50, 63] mm. Table 1 shows the computed x_{50} methods of the GT, SD and our methods with the x_{50} values that are in the range of the GT in bold. As a conversion factor to from pixel to metres, we apply a scale of 40 px = 1 cm. Split-Desktop shows good results especially when all the fragment sizes are present in the image, e.g. **Dist 1**, **Dist 2**, but has problems with segmentation of the background when only a few larger fragments are present (**Dist 3**). Our method estimates the x_{50} value too large for **Dist 2** and **Dist 3** since some small fragments are missing in the segmentation, which in turn increases the x_{50} value. Especially in the region 3.5 mm our method does not detect many fragments and also the multi-scale approach does not solve this problem. There are two main reasons for this problem, first, the small stones were never annotated

and are not in the training set, second, in the multi-scale approach, up-scaling smooths the image, which in turn removes boundary indicators the network requires.

Figure 7 depicts the qualitative results of the MSELL method and Split-Desktop. Regarding our method, the results clearly indicate that the multi-scale approaches are more accurate than the single scale approaches and also that the ellipse fitting is superior to the circle approximation. The single scale methods only perform well for **Dist 3** since this distribution contains solely large fragments. In the difficult dust and boulder **Dist 4** the single scale approaches completely fail as they are not able to segment the small fragments. For **Dist 1** and **Dist 3**, Split-Desktop over-segments the fragments and interprets the background as large fragments. This behavior is not consistent since the background is mostly filtered in **Dist 2**. In contrast, our method successfully distinguishes between back- and foreground. The dust and boulder distribution **Dist 4** is probably the most challenging in the experiments since it contains many small fragments and only a few large ones. Split Desktop regularly merges smaller fragments into larger ones. **Dist 4** shows that our method can successfully detect many individual fragments even if their sizes differ by one or two orders of magnitude.

Figure 9 shows the computed CDF for the distributions **Dist 3** and **Dist 4** of all the methods. For **Dist 3** all methods show good results, while for the difficult **Dist 4** our MSELLipse method is closest to the ground truth, while SD estimates a wrong, uniform distribution.

4.2 Evaluation on Drone Recordings

In this section we present qualitative result on UAV images from a real world open pit mine after a blast at El Ajibe, Spain. We run MSELLipse method on UAV recordings from 4 different flights, two at a height of 16 m and two at a height of 30 m. Since one UAV image has a resolution of 42 MP, for this evaluation we crop patches with a resolution of [2048 × 2048] and again run on three different scale levels [1024 × 1024], [512 × 512]

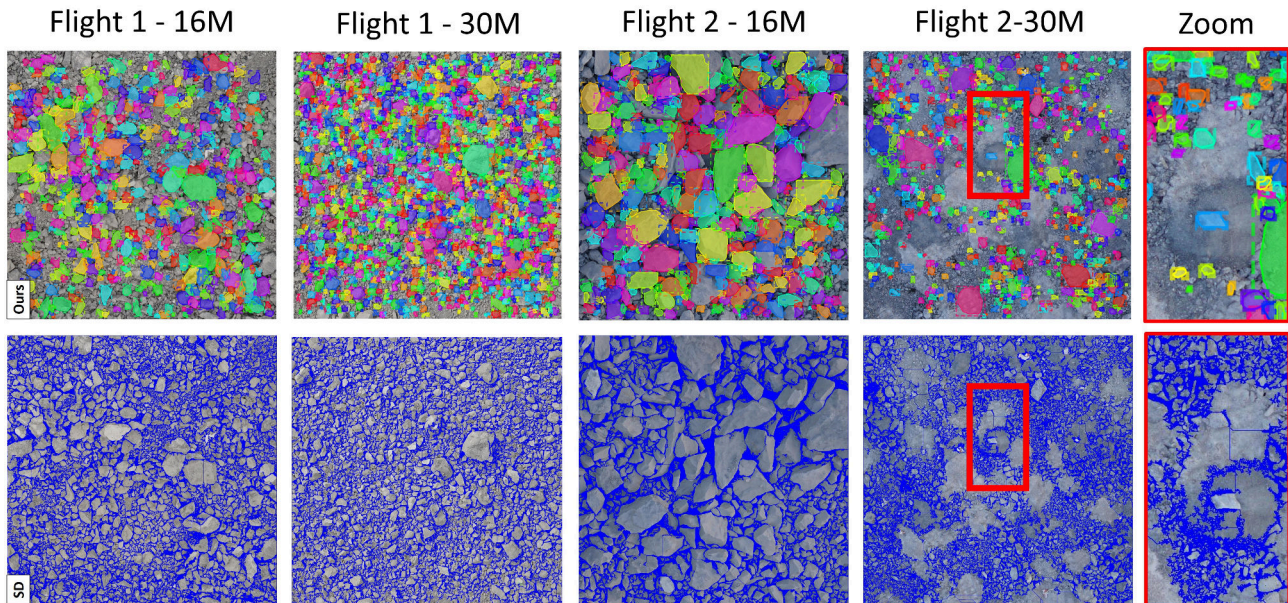


Figure 8. Qualitative results of four UAV flights after a real-world mine blast at 16 m and 30 m. Our MSELLipse method compared to Split-Desktop. While both work well on the first three flights, especially for Flight 2 at 30 m SD misclassifies the background as fragments.

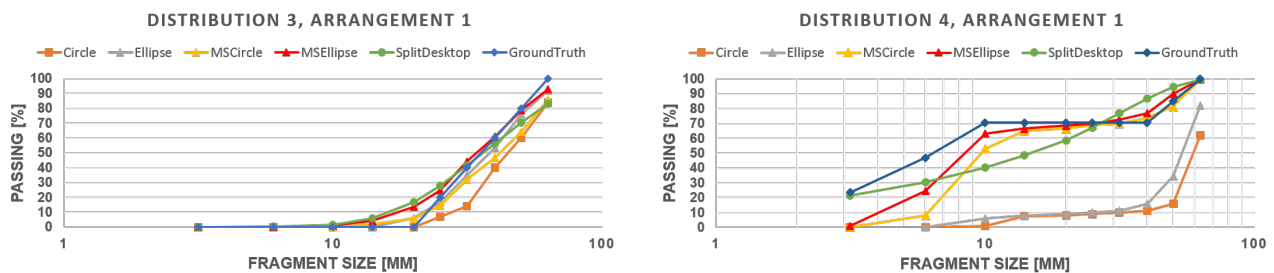


Figure 9. The CDF the distributions **Dist 3** and **Dist 4** of all the methods. While all methods show promising results for **Dist 3**, our MSELLipse method is closest to the GT for **Dist 4**.

and $[256 \times 256]$. Figure 8 shows the qualitative comparison to Split-Desktop.

While both systems show good results for Flight 1 and Flight 2 - 16m, the advantage of our method becomes apparent for Flight 2 - 30 m. Fig. 8 and the zoom clearly show that Split-Desktop interprets the free areas as large fragments. In contrast, our method is able to successfully distinguish between fragments and background.

The UAV experiments indicate that the lab experiments at a smaller scale indeed can be used to generate training data. Our method has neither seen any outdoor nor UAV images during training but the experiments demonstrate that it generalizes well to outdoor scenes. Also note that our method generalizes to other cameras, since the camera in the lab setup is different from the one in the UAV.

5. CONCLUSION

In this work, we presented a novel machine-learning based muck pile characterization method. While many commercially available tools rely on heavy user interaction and require experience, our method works fully automatically. We trained our method

on a lab dataset and demonstrated that it generalizes well to previously unseen lab scenes. Further, we show that our method also works very well in UAV recordings taken after a real-world mining blast, which additionally demonstrates its generalization capabilities.

The next steps are the annotation of smaller fragments in the lab dataset and also the annotation of the UAV recordings to improve the accuracy. We also want to address the challenge of partially occluded fragments. Since we have a complete segmentation of each fragment, we want to analyze other properties such as the shape of the individual fragments (roundness or ratios between the axes).

ACKNOWLEDGEMENTS

This work has been supported from the European Unions Horizon 2020 research and innovation program under grant agreement No 730294 ("Sustainable Low Impact Mining solution for exploitation of small mineral deposits based on advanced rock blasting and environmental technologies", SLIM²).

²<https://www.slim-project.eu/>

REFERENCES

- Dai, J., He, K. and Sun, J., 2016. Instance-aware semantic segmentation via multi-task network cascades. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 3150–3158.
- Dutta, A., Gupta, A. and Zissermann, A., 2016. VGG image annotator (VIA). <http://www.robots.ox.ac.uk/~tevagg/software/via/>. Version: 2.0.1, Accessed: 29.03.2019.
- Fitzgibbon, A., Pilu, M. and Fisher, R. B., 1999. Direct least square fitting of ellipses. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 21(5), pp. 476–480.
- Girshick, R., 2015. Fast r-cnn. In: *International Conference on Computer Vision (ICCV)*, IEEE, pp. 1440–1448.
- Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 580–587.
- He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In: *International Conference on Computer Vision (ICCV)*, IEEE, pp. 2980–2988.
- He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 770–778.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems (NeurIPS)*, pp. 1097–1105.
- Latham, J.-P., Kemeny, J., Maerz, N., Noy, M., Schleifer, J. and Tose, S., 2003. A blind comparison between results of four image analysis systems using a photo-library of piles of sieved fragments. *Fragblast* 7(2), pp. 105–132.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D., 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1(4), pp. 541–551.
- Lin, T.-Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B. and Belongie, S. J., 2017. Feature pyramid networks for object detection. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 2117–2125.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In: *European Conference on Computer Vision (ECCV)*, Springer, pp. 740–755.
- Maerz, N. H. and Zhou, W., 2000. Calibration of optical digital fragmentation measuring systems. *Fragblast* 4(2), pp. 126–138.
- Pinheiro, P. O., Collobert, R. and Dollár, P., 2015. Learning to segment object candidates. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1990–1998.
- Pinheiro, P. O., Lin, T.-Y., Collobert, R. and Dollár, P., 2016. Learning to refine object segments. In: *European Conference on Computer Vision (ECCV)*, Springer, pp. 75–91.
- Sanchidrián, J. A., Segarra, P., Ouchterlony, F. and López, L. M., 2009. On the accuracy of fragment size measurement by image analysis in combination with some distribution functions. *Rock Mechanics and Rock Engineering* 42(1), pp. 95–116.
- Schubert, H., 1989. *Kennzeichnung von Körnerkollektiven, Kennzeichnung von Aufbereitungs- und Trennerfolg, Zerkleinerung, Klassierung. Aufbereitung fester mineralischer Rohstoffe / von Heinrich Schubert, Vol. 1, 4., stark überarb. Aufl. edn*, Dt. Verl. für Grundstoffindustrie, Leipzig.
- Split Engineering LLC, 2016. Split-desktop 3.0 manual.
- Suzuki, S. et al., 1985. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* 30(1), pp. 32–46.
- Thurley, M., 2014. Measuring the visible particles for automated online particle size distribution estimation. In: *International Mineral Processing Congress (IMPC)*.
- Tscharf, A., Schenk, F., Mayer, G., Fraundorfer, F. and Bischof, H., 2018. Uav-anwendungen im bergbau - modellversuche als beitrage zur photogrammetrischen forschung und entwicklung. In: *19. Geokinematischer Tag des Instituts für Markscheidewesen und Geodäsie, TU Bergakademie Freiberg*.
- Yi Li, Haozhi Qi, J. D. X. J. and Wei, Y., 2017. Fully convolutional instance-aware semantic segmentation. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 2359–2367.